STMDF: An Effective Approach for Malicious Domain Detection through Dynamic Spatial-Temporal Analysis

Hongwu Li[1,2,3] , JianQiang Li[4], Xingyu Fu*[1,2,3], DongZheng Jia[4], Yujia Zhu[1,2,3],

Han Wang[1], Qingyun Liu[1,2,3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] National Engineering Laboratory of Information Security Technologies, Beijing, China
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[4] National Computer network Emergency Response technical Team/Coordination Center
fuxingyu@iie.ac.cn

Abstract. The Internet is widely used for network attacks, such as phishing, fraud, gambling, the spread of malware, and botnets. Domains play a crucial role in attackers' network communication due to their low cost and flexibility. Attackers frequently change or transfer malicious domains to evade detection, making it challenging to capture complete associations between domains and related resources. The inherent relationships among domains are difficult to forge, for instance, stable connections exist between domain operators from the same organization or between domains providing similar services. Recent research has employed graph learning techniques, including bipartite graphs, homogeneous graphs, and heterogeneous graphs, to integrate domain attributes and association information for uncovering implicit relationships between domains. However, approaches based on bipartite or homogeneous graphs have limited association information, while methods based on heterogeneous graphs require expert knowledge to design meta-paths and overlook the heterophilic interactions of the domain association graph, where two associated domains may not belong to the same label type. Furthermore, domains and related resources are dynamic, with attributes and associations changing over time. Previous methods have failed to consider the spatiotemporal characteristics. In summary, malicious domain identification techniques require reduced reliance on expert knowledge, consideration of the heterogeneity in graph networks, and attention to the spatio and temporal dynamics of domains and associated resources. In this paper, we propose a novel STMDF model for detecting malicious domains, which utilizes RNN and attention modules to learn temporal information, addressing the complex challenges in malicious domain identification. To validate the effectiveness of our approach, we conduct comprehensive comparisons with various existing detection models, demonstrating the superiority of our method.

Keywords: Spatial-temporal Snapshot Graph Learning, Attention Mechanism, Malicious Domain Identification.

1 Introduction

Due to the low cost and flexibility associated with domains, they often play a crucial role in various types of cybercrimes. One of the most effective and promising approaches is to analyze domain system data for the detection of malicious activities[1]. Attackers exhibit characteristics such as strong anti-detection capabilities. Nonetheless, the relationships behind domains are difficult to forge. The current techniques for identifying malicious domains can be broadly classified into two categories: feature-based methods and association-based methods. Feature-based malicious detection methods extract features closely related to malicious domain [2-5]. In order to evade detection, Attackers can easily mimic the lexical features of benign domains. Such methods treat domain names as independent entities, overlooking the potential interrelationships among domain names.Nowadays methods utilize the inference capabilities of the graph to identify malicious domains employing evasion techniques[6-9]. Methods based on heterogeneous attribute graphs effectively combine domain attribute information and the associations between domain names[10-11]. However, current methods require manual design of meta-paths through expert knowledge[12]. Using heterogeneous graph networks neglect the temporal dynamics of domain names in real-world usage scenarios[13]. However, in the DNS context, the spatial resource associations and temporal behaviors of domain names are closely intertwined. The main contributions of this paper are as follows:

We utilizes a hierarchical attention mechanism to integrate domain attribute information and association information. By aggregating multi-hop neighbors of entities, it mitigates the impact of heterogeneity in the association graph.

We proposes an attention-based spatio-temporal graph neural network approach. Building upon hierarchical attention, it incorporates attention mechanisms and an RNN module to capture temporal information.

Comparative analysis is performed against existing methods such as static homogeneous graphs, dynamic homogeneous graphs, static heterogeneous graphs, and dynamic heterogeneous graphs, demonstrating the superiority of our proposed approach.

2 Motivation

2.1 Spatial context-based correlation

Currently, the geographical attributes of IP addresses have been widely applied and have shown promising results in various industries. The geographical location of an IP address is a projection of the user's real-world spatial characteristic into the cyberspace, reflecting certain regional patterns of network service access. Based on the DataCon2020-DNS malicious domain dataset [14], we extracted the latitude and longitude data of the visiting client IP addresses for 3,410 malicious domains to calculate the

Moran's Index I. The spatial weight matrix used in this section was based on the adjacency of eight neighboring locations. The types of malicious domains used in calculating the Moran's Index included 1,122 gambling domains, 1,009 botnet domains, 788 trojan domains, and 491 APT domains. The results, shown in Table 1, indicate that the Moran's Index values for all three months were greater than 0, with a p-value of 0.01. This suggests a significant spatial clustering characteristic in the distribution of client IP addresses accessing malicious domains.

Table 1. Global Moran's index of client IP accessing malicious domain names.

Month	Moran's Index I	P
March	0.9869	<0.01
April	0.9792	<0.01
May	0.9890	<0.01

The density map of the geographical locations (latitude and longitude) of client IP addresses accessing malicious domains is shown in Figure 1. The darker the color in the figure, the higher the number of users accessing from that region. From Figure 3-1, we can clearly observe clustering patterns, which validate the results of the Moran's Index. Additionally, in Figure 1, we can see that the geographical locations of client IP addresses accessing malicious domains are predominantly positioned to the east of the "Aihui-Tengchong Line" which represents the more developed regions in terms of economy, population, and internet infrastructure. The users accessing malicious domains are more concentrated in coastal provinces, Beijing, and Henan.

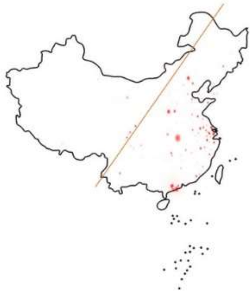

Fig. 1. Geospatial distribution client ip accessing malicious domain.
"Note: This map was created based on the standard map with survey number GS(2016)1555 downloaded from the website of the Ministry of Natural Resources. The base map boundaries have not been modified."

3 Model Design

[15] indicates that many event behaviors exhibit repetitive patterns along the time axis. Therefore, the domain spatiotemporal snapshot graph can effectively utilize the

temporal and spatial information of domains. The first step in constructing the domain association spatiotemporal snapshot graph involves entity extraction. In this study, the domain association spatiotemporal snapshot graph focuses on the DNS resolution process and the entities involved in the domain usage process as the primary objects.

3.1 Domain entity relation extraction

Malicious domains often utilize CDNs to map domain names to IP addresses, evading direct resource connections and camouflaging their traffic as benign. Based on the findings in [16], it is likely that client IPs accessing malicious domains will also access similar malicious domains. During malicious activities, attackers often exhibit correlations between their attack control hosts and infected host groups. For instance, attackers may employ overlapping sets of infected host groups at different stages of an attack, resulting in abnormal correlations between domain query records. Consequently, we establish the (client IP) -- [:query] --> (FQDN) relation. Attackers have limited resources [17], as maintaining a large pool of IP resources incurs significant costs. IP resources tend to exhibit clustering. CNAME domains of compliant domains are unlikely to be malicious, and vice versa. Hence, we establish the (FQDN) -- [:resolve] --> (resolved IP) and (FQDN) -- [:cname] --> (FQDN) relations.

In addition to the relationships between domains and client IPs, and resolved IPs, we incorporate various enriched information for a comprehensive understanding of malicious behavior. For example, the domain registrant (represented by the administrative email) and the ISP (Internet Service Provider) of the IP address. Although domain registration information is often unverified by authoritative sources, it can serve as supporting evidence. Thus, we establish the (ISP) -- [:support] --> (resolved IP), (ISP) -- [:support] --> (client IP), and (administrative email) -- [:register] --> (FQDN) relations.

3.2 Domain spatiotemporal snapshot construction

Currently most of these methods focus on static homogeneous or static heterogeneous graphs. In reality, many real-world graph networks are both heterogeneous and dynamic, often involving multiple types of nodes or edges, which can also dynamically change over time. This dynamic variation can be viewed as a long-term sequence learning problem, aiming to effectively capture precise temporal dependencies between outputs and inputs [18].

The spatiotemporal snapshot graph refers to a static list of snapshots that models the dynamics of a heterogeneous network. Each snapshot represents the structure of the graph network during a specific time period. An individual snapshot consists of a heterogeneous graph, comprising different sets of node types and edge types. Specifically, let's assume we have a dynamic heterogeneous graph network G, which can be represented as an ordered list of spatiotemporal snapshots, G = {g1, g2, ..., gt, ..., gT}, where each snapshot g is composed of different sets of node types Vt and different sets of edge types Et, with t denoting the t-th snapshot. In the node set Vt, each node has a unique identifier, and the number of node types can be represented by the set O. The edge set Et contains all types of edges present in the snapshot g, representing the relationships between nodes, and the number of edge types can be represented by the set R. The purpose of the spatiotemporal snapshot graph is to describe the structural changes of the dynamic heterogeneous network over time.

3.3 Node relation learning based on hierarchical attention heterogeneous graph

This section utilizes a node-level attention model to learn the importance weights of each node's neighbors and generates new low-dimensional vector representations by aggregating the features of these key neighbors. By comprehensively considering the features of these multi-hop neighbors, we can more comprehensively capture the relational dependencies between domain entities, thereby improving the accuracy and robustness of the modeling. The hierarchical attention mechanism consists of two main parts: node-level attention and edge-level attention.

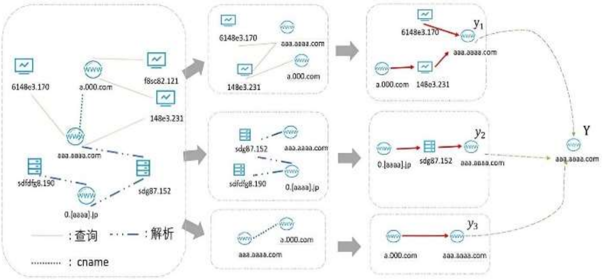

Fig. 2. Hierarchical attention-based malicious domain identification framework

Node-Level Attention： For subgraphs formed by different relationship edges, we employ an attention model [28] for each subgraph with the same edge type. The weight coefficient of each node pair (p, q) in the r-type subgraph is calculated using the following formulas (1) and (2):

$$h'_p = W^{\varphi} \cdot x_p \tag{1}$$

$$\alpha^{\varphi}_{p,q} = \frac{exp\left(\sigma\left(a^{\top}_{\varphi} \cdot [h'_p \| h'_q]\right)\right)}{\sum_{k \in NEI^{\varphi}_p} exp\left(\sigma\left(a^{\top}_{\varphi} \cdot [h'_p \| h'_k]\right)\right)} \tag{2}$$

Subsequently, by aggregating the embedded feature vectors of neighboring nodes [19], we can obtain the final representation of node p under the type edge, as shown in the following formula (3):

$$\mathbf{y}_p^{\varphi} = \sigma\left(\sum_{q \in NEI_p^{\varphi}} \alpha_{pq}^{\varphi} \cdot \mathbf{h}_q^{'}\right) \tag{3}$$

In the domain-resource graph, the entities connected to the domain entity are of different types, such as the client IP entity or the city entity. Therefore, the network exhibits heterophily. In this subsection, we achieve the model's ability to learn heterophily graph networks by learning high-order neighbor information [20]. Aggregating multi-hop neighbor information increases the probability of learning homophily neighbors, which are more informative in graph neural network learning. According to [21], although the immediate neighbors of the current node are predominantly heterophily, the probability of high-order neighbors being homophily increases. The learning method is demonstrated by formula (4):

$$\boldsymbol{y}_p^{\phi} = \|_{k=1}^{K} \left\{F\left(y_p^{\phi}, f\left(\left\{y_p^{\phi}: u \in NEI_i(p)\right\}\right), \ldots\right)\right\} \tag{4}$$

For heterophily networks, after aggregating the neighbor embedding vectors, we do not average the two learned embedding vectors as in GCN, as this would lead to information mixing. In this subsection, we adopt a simple approach of direct concatenation to preserve both parts of the information.

The formula for the multi-head attention mechanism of node p is as follows: given the set of edge types Φ = {1, ..., N}, after calculating the node features through the node-level attention module, we obtain entity feature vectors under N edge types.

Edge-Level Attention: we calculate the normalized weight coefficients β for different edge types by computing the similarity between the mapped vectors [22].

$$\beta_p^{\phi} = \frac{\exp \mathbf{Q}^{\top} \cdot \sigma\left(\mathbf{W} \cdot \mathbf{y}_p^{\phi} + \mathbf{b}\right)}{\sum_1^N \exp\left(\mathbf{Q}^{\top} \cdot \sigma\left(\mathbf{W} \cdot \mathbf{y}_p^{\phi} + \mathbf{b}\right)\right)} \tag{5}$$

in Equation (5) yields the final embedded feature representation of the domain entity.

3.4 Learning temporal information from the domain temporal snapshot graph

We will employ the hierarchical attention heterogeneous model (HAT) from Section 3.3 to learn the static information within individual snapshot graphs. Additionally, we will utilize an RNN module to incorporate the temporal sequence information from the static snapshot sequences. The overall technical architecture is illustrated in Figure 6.

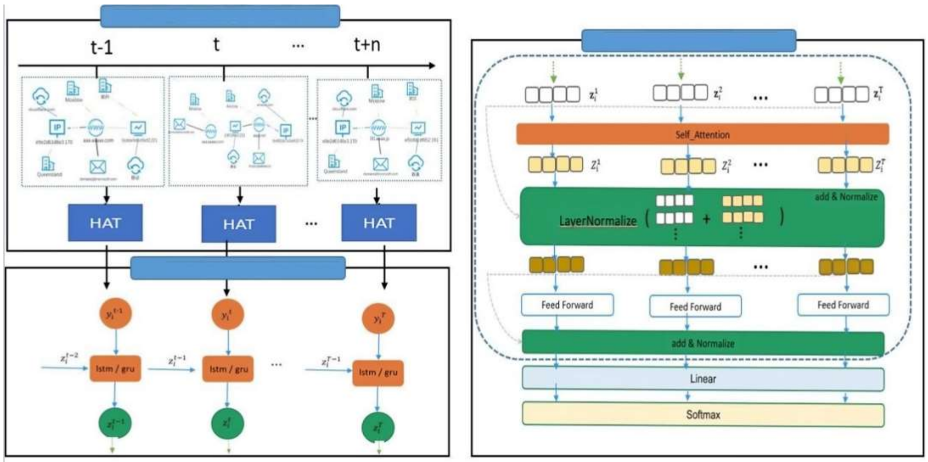

Fig. 3. Architecture diagram of malicious domain identification model based on spatio-temporal snapshot graph

When constructing the malicious domain identification model, we can generate a series of domain-related snapshots on a daily basis (Δt=1, 2, ...). Within each snapshot, we will apply the hierarchical attention mechanism proposed in Section 3.3 to integrate domain attribute information, association information, and spatial information. To capture the temporal changes of domain entities, we will also model the domain entities using a recurrent neural network (RNN) model and an attention mechanism module.

Firstly, our spatio-temporal graph can be extracted as a set of ordered snapshots, denoted as G = {g1, g2, ..., gt, ..., gT} [22], where T is the number of consecutive snapshots. Here, gt = {Vt, Et} represents the t-th snapshot, where Vt is the set of nodes and Et is the set of edges. After the learning process with the model in Section 3.3, as shown in Equation 6, we can obtain the embedded features of domain entity nodes across all temporal snapshot graphs. Here, t represents the t-th snapshot, represents the number of nodes in the t-th snapshot, and F represents the node feature dimension.

$$\left\{\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^T, \mathbf{y}^t \in \mathbb{g}^{|V^t|\times F}\right\} \tag{6}$$

In this chapter, we utilize RNNs to learn the evolution patterns between consecutive snapshots. In this approach, we employ two widely used variants of RNNs, namely LSTM[23] and GRU[24]. The output of the RNN module can be seen as representing the $\{\, \mathbf{z}_i^1, \mathbf{z}_i^2, \ldots, \mathbf{z}_i^T, \mathbf{z}_i^t \in \mathrm{R}^d\}$ node i in the t-th snapshot. We further incorporate a layer of temporal attention module to capture the evolutionary patterns on the dynamic network.

Temporal Attention Module: The node representation from the previous steps is taken as the input. Firstly, based on the attention calculation formula (7) [25], we map the input to different feature spaces to obtain Q, K, and V.

$$Z_i = Attention(Q_i, K_i, V_i) = \text{softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i \quad (7)$$

Then, the output is added to a block Y. Since the input features of the temporal domain entity sequences are multi-dimensional vectors, analyzing each dimension separately is meaningless. Therefore, LayerNorm is introduced here for normalization. The advantage of Layer Normalization is that it normalizes independently across different samples, making the representation of each sample more stable. After normalization, the entities pass through a two-layer fully connected layer as shown in Equation (8). This involves a linear transformation, followed by a non-linear transformation using the ReLU function, and then another linear transformation for forward propagation. Subsequently, as shown in Figure 3, the output is added to the residual block again and undergoes LayerNorm normalization.

$$\text{FFN}(Z) = max(0, zW_1 + b_1)W_2 + b_2 \quad (8)$$

After going through the aforementioned steps, we obtain the embedded features of domain entity node across all snapshots. We use the node features from the last snapshot as the input for the subsequent classification objective function.

4 EXPERIMENTS

The DataCon2020-DNS domain dataset [14] was primarily used as our dataset. This dataset covers domain-related active and passive information captured by 360 between March 1, 2020, and May 31, 2020. It encompasses various types of domains, including normal, gamble, apt, trojans, and botnet domains. In addition to the domain itself, the DataCon dataset also provides WHOIS registration information for each domain, which aids in understanding the background and attributes of the domains. We extracted data from March 27th to May 28th, spanning six weeks, to construct resource snapshots. Each snapshot was constructed every seven days (Δt=7). The specific statistics of the snapshots are illustrated in Table 2.

The experimental evaluation in this chapter utilizes commonly used machine learning metrics, Accuracy and F1. The models compared to our proposed model in this chapter are described as follows: Fanci[26], Deepdom [12], MDFAKG(proposed in Section 3.3), CAW-N [27], NP-GLM [28](metapath2vec and RNN models), STMDF-G(RNN component employs GRU), STMDF-L(RNN component employs LSTM).

Table 2. Statistical information of domain resource snapshots.

	Date	**Number of Nodes in Snapshot**	**Number of Edges in Snapshot**
1	2020/0327~2020/0402	34k	58k

2	2020/0403~2020/0409	29k	53k
3	2020/0410~2020/0416	28k	51k
4	2020/0417~2020/0423	36k	57k
5	2020/0424~2020/0430	36k	57k
6	2020/0501~2020/0507	28k	50k

4.1 Model Parameter Settings

In order to compare the performance with the Fanci system, we constructed a random forest model based on the RandomForest model used in the original Fanci paper, using the scikit-learn library. During the validation process, we employed ten-fold cross-validation for testing the model. This involved dividing the labeled dataset into ten smaller datasets, where the model was trained on nine of them and tested on one. This process was repeated ten times, and the average evaluation metrics such as Accuracy and F1-Score were used as the testing results for Fanci.

For the Deepdom model and the proposed model in this chapter, we implemented them based on the torch-geometric library [29]. For the MDFAKG model and CAW-N and STMDF models, which require selecting the number of neighbors, the upper limit of neighbor nodes per layer was set to 25. The learning rate for the classifier was set to 0.005, and the regularization parameter was set to 0.001. The final node dimension for the output feature vector of domain entities was set to 64, the number of attention heads was set to 8, the dropout between network layers was set to 0.5, and the number of hidden layers was set to 4. Stochastic Gradient Descent (SGD) and the Adam optimizer were used for updating and optimization, with the first-moment estimate β1 set to 0.9 and the second-moment estimate β2 set to 0.98.

4.2 Performance comparison

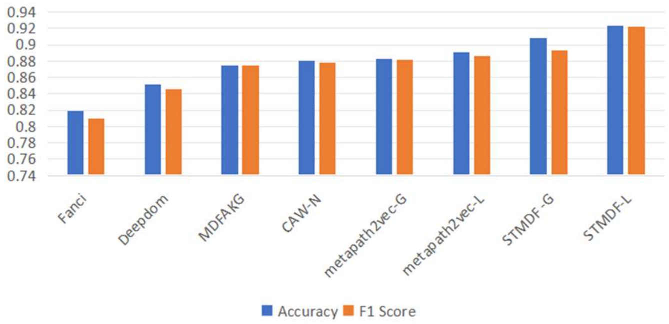

Fig. 4. Performance comparison graph

Comparing with the FANCI method, which achieved scores of 0.8197 and 0.8102 in the metrics, it can be observed that considering the hidden associations between domain names significantly improves the classification performance of the model. The Deepdom model achieved scores of 0.8516 and 0.8462 in the metrics. Furthermore, compared to the Deepdom model, our proposed method achieves good results without the need for constructing meta-paths. The proposed model exhibits better generalization and does not require expert knowledge for selecting meta-paths.

The STMDF model proposed in this chapter outperforms the CAW-N model and the classical dynamic heterogeneous network model NP-GLM in both Accuracy and F1-score metrics. The highest Accuracy and F1-score achieved by STMDF-L are 0.9233 and 0.9219, respectively, outperforming the CAW-N baseline with scores of 0.8806 and 0.8786, metapath2vec-G with scores of 0.8824 and 0.8817, metapath2vec-L with scores of 0.8911 and 0.8867, and STMDF-G with GRU utilizing scores of 0.9087 and 0.8932. Comparing with the CAW-N model reveals that models considering dynamic heterogeneity learning achieve better detection performance, and our model exhibits the best performance, demonstrating the superiority of our method in modeling dynamic changing information.

Comparison of Snapshot Granularity's Impact： Describing the dynamic network as an ordered list of snapshots, Table 2 reveals that the duration of each snapshot influences the total number of snapshots. For example, in our experimental dataset, if the snapshot duration (Δ t) is 21 days, 10 days, and 7 days, there will be 2, 4, and 6 snapshots respectively.

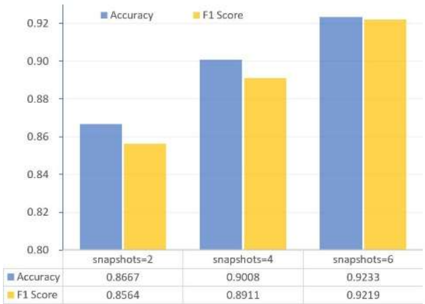

Fig. 5. Impact of snapshot granularity on model results graph

From the comparison results, it can be observed that finer snapshot granularity, meaning a higher number of snapshots, leads to improved performance. When there are only 2 snapshots, the model achieves an Accuracy of 0.8667 and an F1 Score of 0.8564, which are lower than the metrics of 0.9008 and 0.8911 achieved by the model with 4 snapshots. When the number of snapshots in the domain temporal snapshot graph is 6, the model achieves the best performance in terms of evaluation metrics, with an Accuracy of 0.9233 and an F1 Score of 0.9219. The reason behind this improvement is that finer-grained snapshots are more effective in capturing dynamic temporal changes.

Acknowledgement

This work is supoorted by the Research Program of Institute of Information Engineering,CAS (Grant No. E3H0061101)

References

1. Zhauniarovich Y, Khalil I, Yu T, et al. A survey on malicious domains detection through DNS data analysis[J]. ACM Computing Surveys (CSUR), 2018, 51(4): 1-36.
2. Bilge L, Kirda E, Kruegel C, et al. Exposure: Finding malicious domains using passive DNS analysis[C]//Ndss. 2011: 1-17.
3. Grill M, Nikolaev I, Valeros V, et al. Detecting DGA malware using NetFlow[C]//2015 IFip/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015: 1304-1309.
4. Zhao H, Chen Z, Yan R. Malicious domain names detection algorithm based on statistical features of URLs[C]//2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, 2022: 11-16.
5. Ahmed J, Gharakheili H H, Raza Q, et al. Real-time detection of DNS exfiltration and tunneling from enterprise networks[C]//2019 IFip/IEEE Symposium on Integrated Network and Service Management (IM). IEEE, 2019: 649-653.
6. Zou F, Zhang S, Rao W, et al. Detecting malware based on DNS graph mining[J]. International Journal of Distributed Sensor Networks, 2015, 11(10): 102687.
7. Khalil I, Yu T, Guan B. Discovering malicious domains through passive DNS data graph analysis[C]//Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. 2016: 663-674.
8. Lei K, Fu Q, Ni J, et al. Detecting malicious domains with behavioral modeling and graph embedding[C]//2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019: 601-611.
9. Tran H, Nguyen A, Vo P, et al. DNS graph mining for malicious domain detection[C]//2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017: 4680-4685.
10. Sun X, Tong M, Yang J, et al. {HinDom}: A robust malicious domain detection system based on heterogeneous information network with transductive classification[C]//22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019). 2019: 399-412.
11. Sun X, Yang J, Wang Z, et al. Hgdom: Heterogeneous graph convolutional networks for malicious domain detection[C]//NOMS 2020-2020 IEEE/IFip Network Operations and Management Symposium. IEEE, 2020: 1-9.
12. Sun X, Wang Z, Yang J, et al. Deepdom: Malicious domain detection with scalable and heterogeneous graph convolutional networks[J]. Computers & Security, 2020, 99: 102057.
13. Wang Q, Dong C, Jian S, et al. HANDOM: Heterogeneous Attention Network Model for Malicious Domain Detection[J]. Computers & Security, 2023, 125: 103059.
14. 360. DataCon2020-DNS malicious domain dataset [EB/OL]. [2024-02-10]. https://Datacon.qianxin.com/opendata/openpage?resourcesId=1.

15. Zhu C, Chen M, Fan C, et al. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(5): 4732-4740.
16. Rahbarinia B, Perdisci R, Antonakakis M. Efficient and accurate behavior-based tracking of malware-control domains in large ISP networks[J]. ACM Transactions on Privacy and Security (TOPS), 2016, 19(2): 1-31.
17. Sun X, Tong M, Yang J, et al. {HinDom}: A robust malicious domain detection system based on heterogeneous information network with transductive classification[C]//22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019). 2019: 399-412.
18. Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(12): 11106-11115.
19. Wang X, Ji H, Shi C, et al. Heterogeneous graph attention network[C]//The world wide web conference. 2019: 2022-2032.
20. Zhu J, Yan Y, Zhao L, et al. Beyond homophily in graph n neural networks: Current limitations and effective designs[J]. Advances in neural information processing systems, 2020, 33: 7793-7804.
21. Pei H, Wei B, Kevin C, et al. Geom-GCN: Geometric Graph Convolutional Networks[C]// International Conference on Learning Representations.ICLR,2020.
22. Xue H, Yang L, Jiang W, et al. Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal rnn[C]//Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I. Springer International Publishing, 2021: 282-298.
23. Shi X, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[J]. Advances in neural information processing systems, 2015, 28.
24. Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.
25. aswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
26. Waksman A, Suozzo M, Sethumadhavan S. FANCI: identification of stealthy malicious logic using boolean functional analysis[C]//Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013: 697-708.
27. Yan W, Yen C, Yun L, et al. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks[C]// International Conference on Learning Representations. ICLR, 2021.
28. Sajadmanesh S, Bazargani S, Zhang J, et al. Continuous-time relationship prediction in dynamic heterogeneous information networks[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2019, 13(4): 1-31.
29. Rozemberczki B, Scherer P, He Y, et al. Pytorch geometric temporal: Spatiotemporal signal processing with neural machine learning models[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 4564-4573.